

5

10

## Distributed Quality-of-Service Management System

15

BACKGROUND OF THE INVENTION

Mobile devices might comprise applications, protocol stacks as  
20 well as a modem unit adapted for establishing a wireless connection with a mobile communication network, whereby all said entities may run on one processing system. The modem unit might as well be installed on a different processing resource than the applications. Besides that, for the modem unit, a  
25 first operating system might be used, and for the applications, another operating system might be employed. It is expected that in the future, more and more user terminals will be available that do not comprise a proprietary modem unit. Instead, said user terminals will comprise a physical interface for establishing a connection between said user terminal  
30 and an external modem unit, whereby the external modem unit will be responsible for setting up a wireless connection with the mobile communication network. Besides that, phones will be available where an additional processing resource and/or an  
35 additional operating system like e.g. Symbian are provided for

CONFIRMATION COPY

running the applications. Such phones are often called smart-phones.

To ensure QoS in a system and to optimize packet flows it is  
5 necessary to have a connection between the different protocol  
layers, especially if wireless links are used. Higher layer  
protocols and applications have to be tuned according to the  
underlying (wireless) link and according to the actual status  
of the network connection (bandwidth delay, Bit Error Rate  
10 (BER)...). Therefore measurements and parameters from the lower  
layers (GPRS stack, UMTS stack, ...) have to be transported to a  
higher layer QoS management system and/or to higher layer pro-  
tocol / application optimizers. These adjust higher layer pro-  
tocols, applications and packet flows by using these measure-  
15 ments and parameters.

Whereas in an embedded system it is relatively easy to realize  
this vertical connection between the different layers, it be-  
comes more difficult in a distributed system. In a distributed  
20 system the modem contains the information, which is required  
by the QoS Management system residing on the application unit.  
Distributed systems are (other than embedded systems) well  
standardized and defined by well-known specifications. But the  
transfer of measurements and parameters from a (wireless) mo-  
25 dem to the application unit is not part of these specifica-  
tions.

#### SUMMARY OF THE INVENTION

30

It is an object of the invention to improve flow control in a  
distributed user equipment comprising a modem unit and at  
least one application unit.

It is a further object of the invention to transport measurement data from a (wireless) modem to an application unit with the following key features:

- 5   - The transfer of the data should be independent of the interface (serial, IRDA, Bluetooth, ...) between the application unit and the modem.
- No changes to the standard AT command set (necessary for communication between an application unit, e.g. a computer, and a modem) should be required.
- 10   - No changes to the modem driver necessary should be required.
- No changes to the operating system drivers should be required.
- No changes to the PPP (point to point connection) stack should be required.

15

These objects of the invention are solved by the independent claims. Preferred embodiments are shown by the dependent claims.

20

In what follows, the following definitions of terms will be applied:

Distributed System:

- 25   Distributed system is a combination of an application unit and at least one modem. Examples for a distributed system are smartphones and notebooks (laptops) combined with at least one modem.

- 30   Embedded System:

In an embedded system the processor (and the DSP) is (are) used for the applications and higher layer protocols and the (wireless) communication stacks (e.g. GPRS). The classical embedded system in mobile communication is a mobile phone.

35

### Application Unit:

In a distributed system the application unit is the hard- and software environment where the applications run. It has at least a processor, memory and a operating system. The

5 TCP/UDP/IP stack runs also on the application unit, however the stacks for (mobile) communications (i.e. GSM/GPRS) run on an (external) modem and therefore not on the processor of the application unit. The application unit can access different types of modems for communication purposes.

10

### Modem:

A modem is system containing hardware and software (mainly the communication stack) which is used for (mobile) communications. In a distributed system, the modem does not run (execute) any applications.

15

### Interface:

The Interface connects the modem with the application unit. Interfaces can, e.g., be a USB interface, a serial interface, an IRDA interface, a Bluetooth interface.

20

An application unit according to embodiments of the present invention comprises at least one application, wherein said at least one application is adapted for exchanging data traffic for wireless communication with at least one protocol stack.

25

Said at least one protocol stack is adapted for transferring data traffic between at least one of said applications and at least one physical interface. The at least one physical interface is adapted for transmitting data traffic as well as flow control information between said application unit and a modem unit.

30

The application unit is connected with the modem unit via at least one physical interface, whereby both uplink traffic and downlink traffic related to the applications is transmitted via said at least one physical interface. According to the invention, flow control information may be transmitted via the at least one physical interface. The data traffic and the flow control information might e.g. be transmitted in a multiplexed mode via said at least one physical interface. Alternatively, the flow control information might e.g. be transmitted via different physical interfaces than the data traffic.

In prior art solutions, the application unit has not been aware of the flow parameters on the modem unit, and the modem unit has not been aware of the applications and the flow parameters on the application unit. The embodiments of the present invention allow for an exchange of flow control information between the application unit and the modem unit. On both of the units, said flow control information is helpful for utilizing the available resources and the available bandwidth of the wireless connection as efficiently as possible. Each of the units is informed about the overall system state and may react accordingly. As a result, the system's overall performance is increased. A smooth adaptation of the system's control settings is achieved, and the applications' QoS requirements can be fulfilled to a degree that has not been possible yet.

According to a preferred embodiment of the invention, said at least one physical interface is realized as or comprises at least one serial interface, in particular at least one of a RS232, IrDA, Bluetooth, USB, PCMCIA, UART interface. A serial interface provides sufficient bandwidth for transmitting both application-related data traffic and flow control information between the modem unit and the application unit.

According to a preferred embodiment of the invention, said flow control information comprises at least one of: QoS profiles of said applications, actual parameters indicating the actual state of the data flow on at least one of the application unit and the modem unit, and predicted parameters indicating a future state of the data flow on at least one of the application unit and the modem unit. On the application unit, the applications and their QoS requirements may be known. By transmitting these QoS profiles to the modem unit and adjusting the modem unit's parameters accordingly, the distributed system's overall data flow can be optimised, and the applications' QoS requirements can be fulfilled as far as possible. Furthermore, actual parameters indicating the system's actual flow situation might be collected on at least one of the application unit and the modem unit. In order to optimise the overall data flow in a distributed system, each of the units has to be aware about the data flow on the remote units, because this allows to adjust the parameters of the own unit in accordance with the system's overall data flow. Flow parameters and QoS parameters collected on the modem unit might e.g. be provided, as a part of the flow control information, to the application unit. The settings on the application unit may then be adapted accordingly. Vice versa, the application unit might inform the modem unit about types of data traffic, about buffer fill levels and other flow parameters. The exchange of flow parameters helps to find suitable control settings for improving the overall data flow. Both the modem unit and the application unit get a complete picture, and therefore, the decision-making is improved. By means of prediction methods, parameters indicating a future system state might be derived from the actual flow parameters. Said predictions might e.g. anticipate the occurrence of congestion, of cell reselections, of sudden changes of the available bandwidth, etc. By predicting the system behaviour in the near future, a smooth control of the system's settings is accomplished. In order to inform

the application unit about predictions that have been calculated on the modem unit, said predictions are transmitted, as a part of the flow control information, to the application unit. Vice versa, the application unit might provide its predictions to the modem unit.

According to another preferred embodiment of the invention, said flow control information comprises control settings adapted for controlling the data flow on at least one of the application unit and the modem unit. For example, control settings for the entire system may be generated on the part of the modem unit. Said control settings comprise control settings for entities on the application unit that have to be transmitted, as a part of the flow control information, from the modem unit to the application unit. Whenever control settings for an entity on a remote unit are generated, said control settings have to be transmitted, as a part of the flow control information, via the at least one physical interface.

In a preferred embodiment of the invention, said application unit is adapted for receiving, from said modem unit, at least one of control settings for said applications, control settings for said protocol stacks, control settings for buffers. Said control settings are transmitted as a part of said flow control information.

According to another preferred embodiment of the invention, said application unit is adapted for transmitting to said modem unit at least one of: information about said applications, QoS profiles of said applications, information about the protocols used by said applications, types of data traffic, bandwidth per traffic type, maximum buffer sizes, buffer fill levels. Said information is transmitted as a part of said flow control information.

According to another preferred embodiment, said application unit is adapted for transmitting to said modem unit at least one of: predictions related to cell reselections, predictions related to throughput, predictions related to bit error rate, predictions related to coding scheme, predictions related to one way delay, predictions related to round trip time, wherein said predictions are transmitted as a part of said flow control information. Predictions that have been calculated on the application unit might be provided to the modem unit, because, as an example, settings of the transmission protocol stack might have to be adjusted accordingly. Because of the low transmission delay, the predictions will still be valuable when they arrive at the modem unit. In case the modem unit's processing system is employed for performing calculations, e.g. in order to predict cell reselections, the application units connected to said modem unit might be notified. In both cases, the predictions are transmitted as a part of the flow control information via the at least one physical interface.

According to a preferred embodiment, a set of virtual interfaces is used for setting up, maintaining and terminating one or more data connections via said at least one physical interface. Via the at least one physical interface, a lot of different data streams have to be transmitted, whereby said data streams might have completely different properties, and whereby different priorities might be assigned to said data streams. Some of the data streams might arise from different kinds of applications, others might contain flow control information, etc. Said data streams are provided to different virtual interfaces. This allows to distinguish between said data streams, and to process each of the data streams in accordance with its priority and its specific needs.

According to a preferred embodiment of the invention, a permanent high-priority connection is set up between said applica-

tion unit and said modem unit, with said flow control information being transmitted via said high-priority connection.

When measured data, QoS parameters, predictions and control settings are transmitted between the modem unit and the application unit, the transmission delay must not be too high. Otherwise, the flow control information will already be outdated when it is received. For this reason, a high priority has to be assigned to the flow control channel.

According to a preferred embodiment, said at least one physical interface is adapted for transmitting said data traffic and said flow control information in a multiplexed mode between said application unit and said modem unit. The transmission bandwidth provided by the at least one physical interface has to be shared between various different data streams. By transmitting data of different data streams in a multiplexed mode, a plurality of different data streams may be transmitted at the same time, whereby said data streams will not interfere with each other.

In another preferred embodiment, said application unit comprises at least two physical interfaces, whereby said flow control information is transmitted via different physical interfaces than said data traffic.

According to a preferred embodiment of the invention, the application unit comprises a first communication handler module adapted for coordinating and prioritising data traffic between functional entities of the application unit and the modem unit. The first communication handler may e.g. provide a variety of services that relate to allocating, administrating and terminating connections via the at least one physical interface. It may assign priorities to the data streams that are to be transmitted via the at least one physical interface, in or-

der to make sure that the most important data streams are transmitted even if the available bandwidth is not sufficient to carry all traffic.

5 According to another preferred embodiment of the invention, the application unit comprises a first controller module adapted for receiving at least one of flow control information collected on the application unit and flow control information received via said at least one physical interface, and for de-  
10 riving, from said inputs, control settings in a way that the overall data flow is optimised. Said first controller module might e.g. be provided with flow control information from the application unit, from the modem unit, or from both units. The first controller module might e.g. be aware of the applica-  
15 tions' QoS profiles, of the actual state of the data flow within the user equipment, and of predictions that have been made. The first controller module is responsible for decision-making, i.e. it has to derive control settings from the known information. The task is to adjust arbitrary parameters on the  
20 application unit, on the modem unit, or on both units in a way that the overall data flow on the distributed system is optimised. It is attempted to use the available resources in a way that the requirements of the various types of data traffic can be fulfilled as far as possible.

25 In a preferred embodiment of the invention, said first controller module may act as a primary controller that controls a second controller module on the modem unit. In case there exists a first controller module on the application unit and a  
30 second controller on the modem unit, it might be advantageous to select one of the two controller modules as a primary controller that is responsible for generating the control settings for the entire system, or at least for parts of the entire system. If one central instance is responsible for decid-  
35 ing about the control settings, a coherent and well-

coordinated set of control settings will be generated. It might be advantageous to select the application unit's first controller module as a primary controller, because there might be much more memory and computation power available on the application unit than on the modem unit.

According to another preferred embodiment of the invention, said first controller module may act as a secondary controller that is controlled by a second controller module on the modem unit. The second controller module on the modem unit might as well be selected as a primary controller. In this case, the first controller module on the application unit might act as the second controller module's slave.

Especially if two or more application units are connected with one modem unit, it is advantageous to establish the second controller module on the modem unit as the primary controller.

According to a preferred embodiment of the invention, the application unit comprises at least one protocol optimiser module adapted for accessing the settings of corresponding protocol stacks, preferably in accordance with control settings.

According to a preferred embodiment of the invention, the application unit comprises a first QoS packet processor module adapted for at least one of monitoring and modifying the data traffic between at least one of the protocol stacks and said at least one physical interface. Before data traffic of any one of the applications installed on the application unit may be forwarded, via the at least one physical interface, to the modem unit, said data traffic has to pass the first QoS packet processor. This provides an opportunity for monitoring the data traffic, and unknown types of data traffic may be detected and considered by the QoS management system. Furthermore, the first QoS packet processor module may actively mod-

ify certain data streams by holding back or even discarding data packets. The first QoS packet processor module might receive control settings from a respective primary controller, no matter whether said primary controller is located on the application unit or on the modem unit.

According to a preferred embodiment of the invention, the protocol stacks comprise at least one of WAP, TCP, WTCP, UDP, UDP Lite, RTP/RTCP protocol stacks.

A modem unit for mobile communication according to embodiments of the present invention comprises a broadcast facility adapted for setting up a wireless connection for mobile communication. The modem unit further comprises at least one transmission protocol stack adapted for transferring data traffic between said broadcast facility and at least one physical interface. Said at least one physical interface is adapted for transmitting data traffic as well as flow control information between the modem unit and at least one application unit. The modem unit according to embodiments of this invention allows for an exchange of flow control information between the at least one application unit and the modem unit. In particular, the modem unit might e.g. be informed about the data flow on the application unit, and it might e.g. inform said at least one application unit about its actual status. By exchanging flow control information, the system's overall control is improved. A smooth adaptation of the system's control settings is achieved, and the applications' QoS requirements may be fulfilled to a degree that has not been possible yet.

In a preferred embodiment of the invention, said modem unit is adapted for transmitting to at least one of the application units at least one of: parameters of the air link, signal strength of the wireless connection, parameters of the transmission protocol stack, available bandwidth, maximum buffer

sizes, buffer fill levels, information about PDP contexts, radio resource management information. Said information is transmitted as a part of said flow control information.

5 According to a preferred embodiment of the invention, said modem unit is adapted for receiving, from at least one of the application units, at least one of: information about said applications, QoS profiles of said applications, information about the protocols used by said applications, types of data  
10 traffic, bandwidth per traffic type, maximum buffer sizes, buffer fill levels. Said information is transmitted as a part of said flow control information. Thus, the control settings on the modem unit can e.g. be adjusted to the QoS requirements of the data streams that have to be transmitted.

15 According to a preferred embodiment of the invention, said modem unit is adapted for receiving, from at least one of the application units, at least one of: control settings for the transmission protocol stack, control settings for PDP con-  
20 texts, control settings for buffers. Said control settings are transmitted as a part of said flow control information.

In a preferred embodiment of the invention, the applications' QoS profiles are taken into account by at least one of: set-  
25 ting PDP contexts, setting PDP subcontexts, setting or modifying GGSN filter rules. A PDP (Packet Data Protocol) context allows to define the transmission properties for a certain kind of data traffic. By means of a PDP context, it is possible to specify the transmission parameters of the transmission  
30 protocol stack as well as the transmission properties of the wireless link up to the mobile communication network's GGSN (Gateway GPRS Support Node).

According to a preferred embodiment of the invention, the mo-  
35 dem unit comprises a second QoS packet processor module

adapted for at least one of monitoring and modifying the data traffic between said at least one physical interface and the transmission protocol stack. In addition to data traffic generated by applications on the application unit, the second QoS packet processor module might also detect traffic that arises from applications on the modem unit, and it may identify bandwidths and QoS requirements of said data traffic. Control settings can then be chosen in a way that the respective requirements of the applications (including those on the modem unit) are fulfilled as far as possible.

According to a preferred embodiment of the invention, the modem unit comprises a command interpreter adapted for receiving and processing at least one of messages and commands issued by at least one of the application units, in particular for receiving and processing initialisation messages. The command interpreter monitors the data traffic that is transmitted to the modem unit via the at least one physical interface. If a certain command is detected within said data traffic, said command will be interpreted, and the corresponding actions will be carried out. The services provided by the command interpreter are especially useful in case an application on any one of the application units intends to transmit data via the air interface though the entities on the modem unit have not been initialised yet. In this case, the respective application unit sends initialisation messages via the at least one physical interface. On the modem unit, said initialisation messages are detected by the command interpreter. The command interpreter might induce an initialisation of the required entities, and then, the modem can start transmitting data via the wireless link. Thus, the command interpreter allows for a remote initialisation of entities on the modem unit.

In a preferred embodiment of the invention, said transmission protocol stack is a stack for at least one of: GPRS/GSM,

GPRS/EDGE, CDMA, UMTS, wireless LAN, Bluetooth, HiperLan. The invention is in no way limited to any of said transmission protocols, though.

5 A user equipment might comprise at least one application unit and a modem unit as described above, whereby the units are connected via at least one physical interface. Data traffic as well as flow control information is transmitted, via said at least one physical interface, between at least one of the ap-  
10 plication units and said modem unit. By exchanging flow control information between the modem unit and the one or more application units, said units can be brought together more closely. Measured data and predictions are collected from various parts of the system, and the application units might  
15 e.g. inform the modem unit about the QoS profiles of the various applications. From these inputs, control settings for the entire system are derived, and said control settings are distributed to the various entities. As a result, the modem unit and the one or more application units merge and work together  
20 as one system.

According to a preferred embodiment of the invention, said modem unit and at least one of the application units are implemented as one embedded mobile device, preferably as a smart-  
25 phones. Within the embedded device, the application unit and the modem unit might e.g. run on different processing systems. On the modem unit, a first operating system might be employed, while on the application unit, another operating system like e.g. Symbian might be used that is better suited for the re-  
30 spective applications.

According to another preferred embodiment, said modem unit is implemented as a separate device, preferably as a CF card, as a PCMCIA card, or as a part of a mobile phone. According to a  
35 preferred embodiment of the invention, at least one of the ap-

plication units is implemented as a separate device, preferably as a laptop, as a mobile terminal, or as a PDA. A user terminal does not necessarily have to comprise a modem unit. Instead, it might comprise at least one physical interface for establishing a connection with an external modem unit.

The modules according to embodiments of the present invention can be partly or entirely embodied or supported by suitable software, which can be stored on or otherwise provided by any kind of data carrier, and which might be executed in or by any suitable data processing unit.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and many of the attendant advantages of the present invention will be readily appreciated and become better understood by reference to the following detailed description when considering in connection with the accompanied drawings.

Fig. 1 shows an application unit that is part of a user equipment for mobile communication, together with a portion of a modem unit;

Fig. 2 shows a modem unit that is part of a user equipment for mobile communication, together with a portion of the application unit of Fig. 1;

Fig. 3 depicts the protocol layers that can be used for transmitting data in a multiplexed mode via the physical interface;

Fig. 4 shows the structure of the system used for flow control information transfer from a modem to an application unit according to an alternative embodiment;

Fig. 5 shows implementation variants for the application unit collector;

Fig. 6 shows an IP packet for measurement transfer using a proprietary protocol extension; and  
Fig. 7 shows a UDP/IP packet for measurement transfer using a proprietary protocol extension.

5

#### DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

Fig. 1 and Fig. 2 depict a user equipment for mobile communication that comprises an application unit 1 and a modem unit 2. The application unit 1 and the modem unit 2 are connected via a physical interface 3. A plurality of user applications might run on the application unit 1. In the example shown in Fig. 1, the applications comprise e-mail 4, a web-browser 5, DSR (Digital Surveillance Recorder), Push2Talk 6, Video Conference 7, MMS (Multimedia Messaging Service), IM (Instant Messaging), etc. The application unit 1 might further comprise transport protocol stacks with protocol layers like e.g. RTP/RTCP (Real Time Transport Protocol, Real Time Transport Control Protocol), RSVP (Resource ReserVation Protocol), WSP (Wireless Session Protocol), UDP (User Datagram Protocol), UD-PLite, TCP (Transmission Control Protocol), WTCP (Wireless profiled TCP), WAP (Wireless Application Protocol), etc. The transport protocol stacks transform the data payload of the various applications into IP (Internet Protocol) packets. The application-related data, and in particular the IP packets, are exchanged, via the physical interface 3, with the modem unit 2. The physical interface 3 might e.g. be realized according to one of the standards RS232, USB (Universal Serial Bus), Bluetooth, IrDA (Infrared Data Association), PCMCIA, etc. or by means of UARTs (Universal Asynchronous Receiver - Transmitter).

The modem unit 2 is implemented as a separate unit. The modem unit 2 is responsible for establishing and maintaining a wire-

less connection to a mobile communication network. For this purpose, the modem unit 2 comprises at least one transmission protocol stack 8. Said transmission protocol stack 8 might e.g. be a GPRS/GSM stack, or a GPRS/EDGE stack, or a stack for a future transmission protocol such as UMTS. IP packets that are received from the application unit 1 via the physical interface 3 are transferred to the transmission protocol stack 8. Vice versa, data received via the wireless connection is provided to the transmission protocol stack 8 and is routed, via the physical interface 3, to the application unit 1. The modem unit 2 might further comprise internal applications 9 and corresponding protocol stacks adapted for exchanging data traffic with the transmission protocol stack 8.

In the set-up shown in Fig. 1 and Fig. 2, the application unit 1 and the modem unit 2 are realized as separate units. For example, the application unit 1 might run on a first CPU (or on a first DSP), whereby a first operating system is used, while the modem unit 2 might run on a second CPU (or on a second DSP), whereby a second operating system is used. Such a set-up might e.g. be found on a so-called smartphone, whereby a dedicated operating system like e.g. Symbian might be installed on the smartphone's application unit. To the end-user, these devices look the same as mobile phone devices, though the application unit and the modem unit are implemented as separate units. The application unit 1 and the modem unit 2 might as well be implemented on different mobile terminals. For example, a mobile device like a phone might be used as a modem for a second device like e.g. a laptop or a PDA (Personal Digital Assistant), whereby applications like e-mail, web-browser, VoIP client, Video applications etc. are executed on the part of the second device. The mobile phone comprises the modem unit, and the second device acts as an application unit. Alternatively, the modem unit 1 might be a dedicated piece of

hardware like a CF (Compact Flash) card, a PCMCIA card, or the like.

In the following, the structure and functionality of a QoS management system for the above mentioned kind of user equipment will be described. It has to be noted that the invention can also be used if two or more application units are connected to one modem unit. Furthermore, the at least one application unit might be connected with two or more different modem units that support different transmission standards.

The applications that are running on the part of the application unit 1 are provided by different third party companies. Some of said applications, for example e-mail 4 and web-browser 5, might not be aware of the QoS management system.

Other applications, like DSR, Push2Talk 6, Video Conference 7, MMS and IM, might be aware of the QoS management system. For these applications, the external application manager 10 is visible. The applications that are aware of the QoS management system register (11) with the external application manager 10 and forward their respective QoS requirements to the external application manager 10. The applications' QoS requirements are usually specified in terms of QoS classes. In general, four basic QoS classes are used that have been defined by 3GPP (3rd Generation Partnership Project), though other classifications might be used as well.

### 30 Conversational class

Traffic of the conversational class is very delay sensitive, and transfer delay and time variance between packets must stay below a certain value so that the human perception will accept the quality of the link. For traffic of the conversational

class, it is most important that data is delivered in time. The bit error rate (BER) of the data traffic is not that critical. Examples for traffic of the conversational class comprise IP telephony and video telephony. In the example of Fig. 1, data traffic of the application Video Conference 7 belongs to the conversational class.

#### Streaming class

Traffic of the streaming class comprises one way real-time traffic. For traffic of the streaming class, a low transfer delay is not necessarily required, but the delay variation of the real-time data stream should be limited. In Fig. 1, data traffic of the application Push2talk 6 belongs to the streaming class.

#### Interactive class

Traffic of this class might e.g. emerge from an application where a user exchanges data interactively with an opposite party, which might either be another user or a computer system. The response to a request is generally expected within a certain time limit. Although the transfer delay may be higher than in case of traffic of the conversational class, the round trip time (RTT) is a key parameter. Traffic of the interactive class should show a low BER. Examples for this kind of traffic comprise web browsing or Telnet. In the example of Fig. 1, data traffic of the application IM belongs to the interactive class.

#### Background class

For data traffic of the background class, low delay or a short delivery time is not an issue, but the bit error rate (BER) has to be low. Data traffic of this class is usually received by a computer. Email traffic is a typical example for this  
5 kind of traffic. Accordingly, data traffic of the application e-mail 4 in Fig. 1 belongs to the background class.

For at least some of the applications that are aware of the QoS management system, so-called application optimisers might  
10 be installed. In Fig. 1, the applications DSR, Push2Talk 6, Video Conference 7, MMS and IM comprise corresponding application optimisers 12-16. When an application intends to send data, the corresponding application optimiser might adjust the way said data is generated to the overall data flow, to the  
15 available bandwidth, to the properties of the wireless connection, etc. In particular, the application optimisers 12-16 might influence the timing, the packaging of data, and the number of application frames per data packet, whereby the applications' QoS profiles are taken into account. During initialisation, the application optimisers 12-16 register with  
20 the external application manager 10. During operation, the application optimisers 12-16 might receive control information 17 from the external main controller 18, with said control information 17 comprising control settings for the application  
25 optimisers 12-16.

Next, the external application manager 10 initialises (19) the external protocol manager 20. The external protocol manager 20 is informed about the applications that run on the application  
30 unit 1, about the transport protocols used by said applications, and about the respective QoS profiles of the applications' data traffic.

For at least some of the transport protocol stacks, so-called  
35 protocol optimisers might be installed. In Fig. 1, protocol

optimisers 21-25 corresponding to the transport protocol stacks RTP/RTCP, WSP, UDP, UDP-Lite and TCP are shown. The external protocol manager 20 initialises (26) the protocol optimisers 21-25 in accordance with the applications' QoS profiles. Each of the protocol optimisers 21-25 is responsible for dynamically accessing and adjusting the properties of its corresponding transport protocol stack. Said adjustment is performed in accordance with the corresponding applications' QoS requirements, in accordance with the overall data flow, and in accordance with measured or predicted system parameters.

Next, the external protocol manager 20 initialises (27) the external main controller 18 and transfers the control of the protocol optimisers 21-25 to said external main controller 18. Further on, both the protocol optimisers 21-25 and the external protocol manager 20 will receive their respective control settings 28 from the external main controller 18.

During operation, changes of the system's set-up and of the data traffic generated by the applications might occur, and accordingly, a re-initialisation of at least one of the external application manager 10, the application optimisers 12-16, the external protocol manager 20 and the protocol optimisers 21-25 might be necessary. Said re-initialisations can either be induced by the applications themselves or by the external main controller 18.

The application unit 1 and the modem unit 2 communicate via the physical interface 3. In addition to uplink and downlink data traffic related to various applications, also flow control information is transmitted between the application unit 1 and the modem unit 2. Said flow control information might e.g. comprise measured parameters that indicate the actual system state, predictions indicating a future system state, and in-

formation related to the system's configuration. The flow control information might also comprise control settings for entities on the application unit 1 and on the modem unit 2.

5 The different data streams transmitted via the physical interface 3 should be handled separately and in accordance with their respective priorities. On the application unit 1 and on the modem unit 2, a plurality of virtual interfaces 29-32 and 33-36 are provided, and said virtual interfaces may be used  
10 for setting up one or more virtual channels between the application unit 1 and the modem unit 2. According to a first embodiment, a corresponding physical interface is assigned to each of the virtual interfaces in a way that a one-to-one correspondence is established between the virtual interfaces and  
15 the physical interfaces. Alternatively, if only one physical interface is available, or if less physical interfaces than virtual interfaces are available, the data streams may be transmitted in a multiplexed mode. In this embodiment, a multiplexing protocol 37 is implemented both on the application  
20 unit 1 and on the modem unit 2. The multiplexing protocol 37 is adapted for providing a plurality of separate virtual interfaces. The multiplexing protocol 37 allows to transmit a plurality of data streams via different virtual channels in parallel, whereby priorities assigned to said data streams are  
25 considered. Accordingly, the transmission of lower priority traffic may be interrupted by higher priority traffic. The multiplexing is controlled by a mux controller 38 on the application unit 1 and by a mux controller 39 on the modem unit 2. Said mux controllers 38, 39 are responsible for setting up  
30 and tearing down virtual channels between the virtual interfaces.

For transmitting data via the physical interface 3 in a multiplex mode, the PPP (Point-to-Point Protocol) suite might be  
35 used together with some extensions. Said extensions allow for

virtual serial lines and traffic prioritisation. A detailed description of the PPP protocol suite can be found in the document IETF RFC 1661.

- 5 Alternatively, the PPP protocol suite without the above mentioned extensions can be combined with one of the multiplexing protocols defined by ETSI (European Telecommunications Standards Institute). Technical specifications of said multiplexing protocols can be found in ETSI, 3GPP TS 07.0 10 and 27.0
- 10 10. According to a third alternative, the application unit 1 and the modem unit 2 might be linked by an IP connection.

In the following, the initialisation of the interface facility will be described. The external main controller 18 addresses

15 the mux controller 38 and allocates a virtual interface for establishing a Fast QoS Connection between the application unit 1 and the modem unit 2. Next, the external main controller 18 initialises (40) the Comm Handler 41 and the Fast QoS Connection 42. The Comm Handler 41 is responsible for handling

20 the communication between entities of the QoS management system located on the application unit 1 and entities located on the modem unit 2. In case of congestion, the Comm Handler 41 has to decide about the priorities of the various data streams.

25

The Fast QoS Connection 42 is a fast communication channel adapted for transmitting flow control information between the application unit 1 and the modem unit 2. The Fast QoS Connection 42 is permanently set up on one channel of the interface.

30 Flow control information, in particular measured data, control settings and predictions, have to be transmitted with low delay. For this reason, one of the highest available priorities is assigned to the Fast QoS Connection 42.

Next, the application unit's Comm Handler 41 issues commands for starting up a main controller 43 and a Comm Handler 44 on the part of the modem unit 2. Said commands are transmitted, via the Fast QoS Connection 42, to the modem unit 2. There, said commands are detected and interpreted by an AT command interpreter 45. In accordance with said commands, the main controller 43 is initialised (46). Then, the main controller 43 initialises (47) the modem unit's Comm Handler 44. Between the application unit's Comm Handler 41 and the modem unit's Comm Handler 44, a link is established. As soon as this link is available, the Comm Handler 41 will inform the external main controller 18.

The external main controller 18 may now forward flow control information via the data link 48 to the Comm Handler 41. From the Comm Handler 41, the flow control information is transmitted, via the Fast QoS Connection 42, to the Comm Handler 44. The Comm Handler 44 forwards the flow control information via the data link 49 to the main controller 43. In the opposite direction, the main controller 43 may transmit flow control information via the data link 49, the Fast QoS Connection 42 and the data link 48 to the external main controller 18. The external main controller 18 and the main controller 43 may now exchange all kinds of flow control information comprising QoS profiles, measured parameters, statistics, predictions, control settings, etc.

At first, the main controller 43 is installed as a primary controller that controls the external main controller 18. The external main controller 18 acts as a secondary controller (slave). The external main controller 18 and the main controller 43 exchange information about their respective capabilities and about the configuration of the application unit 1 and the modem unit 2. Then, the main controller 43 has to decide whether it is favourable to transfer the primary control of

the QoS management system to the external main controller 18 or not. The CPU of the application unit 1 might have much more resources in terms of processing power and memory, and the modem unit's CPU would be offloaded from some of its tasks. However, if two or more application units are connected to the modem unit, the main controller 43 will most likely continue to act as a primary controller and control the tasks of the external main controllers.

10 The primary main controller is responsible for setting up the whole QoS management system. It has to decide how to distribute the required functionalities to the entities of the distributed QoS management system. Then, the respective entities are initialised accordingly. For example, the primary main  
15 controller has to decide whether a state predictor should be initialised on the application unit 1, on the modem unit 2, or on both of said units. The state predictors might use complex algorithms for deriving the respective predictions, and accordingly, the state predictors will demand a lot of computation power. For this reason, it might be advantageous for the  
20 modem unit 2 to offload some of the computations to an external state predictor running on the application unit 1.

In the set-up of Fig. 1 and Fig. 2, the modem unit 2 comprises  
25 a state predictor 50, and the application unit 1 comprises an external state predictor 51. The state predictor 50 on the modem unit 2 is initialised (52) by the main controller 43.

The state predictor 50 receives (53) measured data and system  
30 parameters. The predictions of the state predictor 50 are derived from measured data and system parameters that indicate the actual state of the system. The state predictor 50 comprises a multitude of different state predictor modules. For example, the state predictor 50 might comprise a state predictor  
35 module 54 adapted for predicting one way delay, round trip

time (RTT) and throughput, a state predictor module 55 adapted for predicting coding scheme and BER (Bit Error Rate), and a state predictor module 56 adapted for predicting cell reselections. During a cell reselection, data transmission is interrupted for a period of time in the order of seconds, and therefore, the main controller 43 should be informed about cell reselections. The predictions of the state predictor 50 are provided (57) to the main controller 43. Similarly, the external state predictor 51 is initialised (58) by the external main controller 18. The external state predictor 51 receives (59) measured data and system parameters. It comprises state predictor modules 60, 61, 62 adapted for deriving a variety of different predictions. Said predictions are provided (63) to the external main controller 18.

Further, the external main controller 18 initialises (64) an external QoS packet processor 65. The external QoS packet processor 65 is responsible for detecting and keeping track of the various types of data traffic between the transport protocol stacks and the interface facility. For this purpose, it monitors both the up-link traffic and the downlink traffic. The external QoS packet processor 65 detects the bandwidths and the QoS profiles of the different types of data traffic.

On the application unit 1, there might exist applications that are not aware of the QoS management system. For example, the applications e-mail 4 and web-browser 5 shown in Fig. 1 might belong to the group of applications that is not aware of the QoS management system. If one of said applications starts sending data traffic, the external QoS packet processor 65 will detect this new kind of data traffic. Whenever a new type of data traffic is detected, the external QoS packet processor 65 will identify this traffic, the bandwidth and QoS profile of said traffic, and it will identify the application that has generated said traffic. In addition to that, the external QoS

packet processor 65 may modify the flow of data packets. For this purpose, the external QoS packet processor 65 may hold back and buffer IP packets of certain data streams, whereby data packets of minor importance may even be discarded. The  
5 external QoS packet processor 65 receives control settings 66 from the external main controller 18 that indicate how the filters and buffers have to be set up.

Furthermore, the external main controller 18 initialises (67)  
10 an external collector 68 on the application unit 1. The external collector 68 is responsible for collecting information from different entities of the application unit 1. For example, the external collector 68 receives (69) information including the types of traffic, the current bandwidth per traffic  
15 type, maximum buffer sizes, current fill levels of various buffers, etc. from the external QoS packet processor 65. Furthermore, the external collector 68 might receive feedback information 70 from the transport protocol stacks, e.g. from the RTP/RTCP protocol stack. The external collector 68 provides  
20 (59, 71) the collected information to the external state predictor 51 and to the external main controller 18. Similarly, the modem unit 2 might comprise a collector 72 that is responsible for collecting information from the entities of the modem unit 2.

25  
In addition to the applications that run on the application unit 1, internal applications 9 and the corresponding protocols might as well be installed on the modem unit 2. The internal applications and protocols indicated in Fig. 2 might  
30 additionally comprise at least one of: an application manager, a protocol manager, application optimisers and protocol optimisers. Said entities are part of the QoS management system. They are initialised (73) by the main controller 43, and they receive control settings 74 from the main controller 43.

35

Besides the external QoS processor 65 on the application unit 1, also the modem unit 2 might comprise a QoS packet processor 75 that is initialised (76) by the main controller 43. The QoS packet processor 75 monitors the uplink and downlink traffic.

5 Besides that, it might modify the data traffic passing through it. Data packets may be buffered before they are forwarded to the transmission protocol stack 8, or they may even be discarded.

10 In particular, the QoS packet processor 75 detects and analyses data traffic arising from the internal applications 9. Information about different types of data traffic and their respective bandwidths is forwarded (77) to the collector 72. The primary main controller, e.g. the main controller 43, processes the information provided by the external QoS packet  
15 processor 65 and by the QoS packet processor 75. Based on this information, the primary main controller decides whether the overall QoS can be improved by setting up another PDP context, a PDP subcontext, or a new filter list for the GGSN (Gateway  
20 GPRS Support Node). PDP contexts and PDP subcontexts allow to define the transmission properties for a certain type of data traffic. The primary main controller might instruct (78) the mobility/radio resource management 79 to set up or modify a PDP context or a PDP subcontext. The parameters of said PDP  
25 contexts and PDP subcontexts are chosen in accordance with the QoS requirements of the respective traffic. When the respective PDP context or PDP subcontext has been activated, the primary main controller will instruct (80) the QoS packet processor 75 to use this PDP context or PDP subcontext for the  
30 further transmission of certain types of data traffic.

The transmission protocol stack 8 might e.g. be a GPRS/GSM stack, a GPRS/EDGE stack, a UMTS stack, or a HiperLan stack, or a WLAN stack. In the future, other transmission protocol  
35 stacks that relate to future transmission protocols might be

employed. In case of a GPRS/GSM or a GPRS/EDGE stack, the stack's uppermost layer is a SNDCP (Sub-Network Dependent Convergence Protocol) layer. In case of a UMTS stack, the uppermost layer is a PDCP (Packet Data Convergence Protocol) layer.

5 The subsequent layer, the LLC (Logical Link Control), is responsible for segmenting the IP packets into data blocks suitable for transmission. For this purpose, the LLC comprises a LLC buffer 81. The data blocks are forwarded to a RLC (Radio Link Control) layer comprising a RLC buffer 82. The data  
10 blocks are provided to the physical layer L1, which is the lowest layer of the transmission protocol stack 8. The main controller 43 may initialise (83) a LLC manager 84 that is part of the QoS management system. The LLC manager 84 may set various parameters of the LLC, delete LLC blocks or reorder  
15 LLC blocks. Similarly, the main controller 43 may initialise (85) a RLC manager 86 that is adapted for accessing the settings of the RLC, and for modifying the RLC data blocks.

The control settings of the transmission protocol stack 8 may  
20 be dynamically adapted (87) by a stack manager 88, which is initialised (89) and controlled (90) by the main controller 43. There exist a variety of possibilities how the stack manager 88 can do that: the stack manager 88 might e.g. influence  
(91) the mobility/radio resource management 79 in a way that a  
25 cell reselection is either initiated or delayed. Furthermore, it might reset the RLC buffer 82 and/or delete selected PDUs (Protocol Data Units) in the RLC buffer 82. Besides that, the stack manager 88 might be involved in administrating PDP contexts and PDP subcontexts. Furthermore, the stack manager 88  
30 might be involved in setting the filter rules of the GGSN in accordance with the QoS requirements of the respective traffic. By setting the RLC mode, the stack manager 88 might specify whether an acknowledged or an unacknowledged mode shall be used for the data transmission, and how the delivery of defective  
35 RLC blocks shall be handled.

The mobility/radio resource management 79 is responsible for the mobility management, for authorization, and for establishing and terminating a wireless connection. It is also responsible for performing cell reselections, i.e. for switching from one base station to an adjacent base station. The mobility/radio resource management is instructed to set up PDP contexts and PDP subcontexts with suitable attributes for all kinds of data traffic as well as filter lists for the GGSN.

After the collector 72 on the part of the modem unit 2 has been initialised (92), it starts collecting information from different entities on the modem unit 2. For example, from the physical layer L1, information relating to the signal power and the available bandwidth of the wireless connection might be obtained (93). The collector 72 might further collect information from the RLC (94), from the LLC (95), from the SNDCP/PDCP (96), from the QoS packet processor 75 (77), and from the internal applications 9 (97). The collected data is provided (53, 98) to the state predictor 50 as well as to the main controller 43. Between the application unit's external collector 68 and the modem unit's collector 72, a direct communication might be established, and collected data might be exchanged via the Fast QoS Connection 42.

During operation, the respective primary controller will receive flow parameters from the application unit's external collector 68, and from the modem unit's collector 72. Furthermore, the respective primary controller is provided with predictions from the state predictor 50 and from the external state predictor 51. The primary controller is responsible for making decisions, and for determining the control settings for the entire system in accordance with predefined strategies. The aim is to smoothly adapt the control settings to the requirements of the various data streams. In case the external

main controller 18 is selected as the primary controller, flow parameters and predictions provided by the collector 72 and the state predictor 50 are transmitted via the Fast QoS connection 42 and the data link 48 to the external main controller 18. Control settings that are destined for the modem unit 2 are transmitted via the data link 48 and the Fast QoS connection 42 to the entities on the modem unit 2. The main controller 43, which acts as a secondary controller, might be responsible for distributing the control settings on the modem unit 2.

In case the main controller 43 is selected as the primary controller, flow parameters and predictions provided by the external collector 68 and the external state predictor 51 are transmitted via the Fast QoS connection 42 and the data link 49 to the main controller 43. Control settings for the application unit 1 are transmitted via the data link 49 and the Fast QoS connection 42 to the entities on the application unit 1. In this case, the external main controller 18 acts as a slave of the main controller 43. Said external main controller 18 might be responsible for distributing the control settings on the application unit 1.

It has to be pointed out that a QoS management system according to the present invention doesn't have to comprise every single one of the modules shown in Fig. 1 and Fig. 2. A QoS management system according to an embodiment of the present invention might as well comprise a subset of the above-mentioned modules.

Fig. 3 shows the layers of the transport protocol stack together with the layers of a multiplexing protocol that is used for transmitting data via the physical interface 3. For example, user data 99 that is part of a real-time data traffic might be generated on the application unit 1. A header 100 for

payload is added to said user data 99, and the obtained payload 101 is forwarded to a corresponding transport protocol stack 102. In Fig. 3, the transport protocol stack 102 comprises protocol layers for RTP, UDP and IP. The transport protocol stack 102 provides IP packets to the interface facility. For setting up a connection between the application unit 1 and the modem unit 2, protocols of the PPP protocol suite might be employed (103, 104) on both units. Alternatively or additionally, services provided by a Comm Handler 105 on the application unit 1 and by a Comm Handler 106 on the modem unit 2 might be utilized.

In order to allow for the transmission of multiple data streams via the physical interface 3, a multiplexing protocol is implemented on the application unit 1 and on the modem unit 2. For example, a multiplexing protocol according to one of the standards 3GPP 27.0 10 or 7.0 10 might be used. On the application unit 1, the multiplexing protocol 107 provides a set of virtual interfaces 108, 109. Accordingly, virtual interfaces 111, 112 are provided by the multiplexing protocol 110 on the modem unit 2. Said virtual interfaces may be used for setting up virtual channels between the application unit 1 and the modem unit 2. Then, IP packets may be transmitted from the application unit 1 to the modem unit 2, and vice versa, via the physical layer 113. In addition to user data, messages and commands might be transmitted via the physical layer 113 between the Comm Handlers 105 and 106. The AT command interpreter 114 is used to set up and modify the virtual interfaces.

#### Alternative embodiment

Fig. 4 shows an alternative structure and the corresponding components used for the transfer of the measurements, i.e. the

flow control data, from the modem to the application unit. The usual abbreviations for protocol stacks for the OSI reference model have been used. L1, e.g., means layer 1 or the physical layer. The modifications to the common model are explained in the following. In particular, the following components are used and shown:

#### Sub-Collector:

The Sub-collector is situated in the modem. It collects all the measurements and parameters from the (wireless) stack which are requested by the main controller (or Decider) as long as they are supported by the stack. It builds the IP packet in which the measurements and parameters are transported to the application unit.

#### Sender:

The sender is situated in the modem too. It is responsible to send the IP packets (which include the measurements) to the application unit. This mechanism is described in more details below.

#### Media Sense:

Media Sense is responsible for detecting which modem is connected currently to the application unit, whether this modem is usable at the moment and which parameters are supported by the modem.

#### State Predictor:

The state predictor is capable of predicting the future development of network related parameters. The predictions are based on measurements from the (wireless) stacks. The SP gets the measurements from the AU collector.

#### Main Controller/ Decider:

The Main Controller (decider) is responsible for decision making. Based on the measurements from the wireless stacks (provided from the AU collector), the predictions (provided by the SP) and the status of the modems (provided by the media sense) it decides which strategies, adaptations and packet flow optimizations should be done in this moment.

#### AU collector:

The AU collector (application unit collector) fetches the IP packets out of the packet flow and extracts the measurements. It also builds the IP packets, which are used to request measurements from the modems (see below).

Depending on the features of the IP stack (especially, whether there is a RAW IP socket or not) three different implementation variants are possible as shown in Fig. 5.

#### Variant A:

The QoS Packet Processor is implemented. In this case the measurements are transported in an IP packet with a proprietary transport protocol extension. All IP packets have to go through the QoS PP and it can therefore easily give the measurement packets to the AU Collector and put the AU Collector packets in the packet flow to the modem.

An example of the proprietary transport protocol extension is given below. Fig. 6 shows an IP packet using the proprietary transport protocol extension using common abbreviations for IP packets.

The packet includes a standard IP (version 4) header. In the protocol field "254" is used (open for experimental use).

The first 4 bits of the extension are reserved for a protocol number, called "fgProt". The mechanism may be used to transport other pieces of information, too. Measurement transport has the protocol number 1 (bit coded).

5

The second 4 bits are reserved for the protocol version. Each protocol may need changes in the future. At the moment, the measurement protocol has Version 1 (bit coded).

10 The payload may include several measurement blocks. The first 8 bits of all blocks are showing the kind of measurement, the second 8 bits are showing the length of this measurement block followed by the measurements itself. Each measurement has its own structure.

15

One example:

For GPRS cell reselection predictions the signal strengths of the serving cell and the neighbor cells are transferred. The measurements are taken in a certain time interval. The measurement string will have the following format (without the spaces, spaces are just included for helping the eyes:

20

"S 71 72 N 71 72 57 86 73 87 78 89"

25 This means:

Serving Cell:	C1	ARFCN 71	Signal strength -72 dbm
Serving Cell:	C2	ARFCN 71	Signal strength -72 dbm
Neighbor Cell:	C2	ARFCN 57	Signal strength -86 dbm
30 Neighbor Cell:	C2	ARFCN 73	Signal strength -87 dbm
Neighbor Cell:	C2	ARFCN 78	Signal strength -89 dbm

C1 and C2 are the signal strengths as specified in the GSM/GPRS specification. ARFCN is the ID number of the Cell.

35

The Type of this measurement is specified in a list.

Coming back to the figure showing the proprietary extension, the following fields will have the following attributes:

5

FgProt: 1 (bit coded)

Vers: 1 (bit coded)

Type of measurement: 6 (coded in Bit, means 256 different measurements possible)

10

Length of measurement: 22 (measurement string needs 22 bytes, see below, coded on Bit, maximal length 256 byte)

Measurement: S7172N7172578673877889 (hex coded, each sign needs one byte)

15

Variant B:

The QoS PP is not implemented. The IP stack provides no RAW IP socket or interface. In this case the UDP protocol is used as transport protocol for the measurements. A special (commonly unused) port is opened and used by the AU collector. The AU collector acts like an independent application. Measurements and requests are packed as a proprietary format in a UDP/IP packet.

25

An example of the proprietary format in a UDP/IP packet is given below. Fig. 7 shows a UDP/IP packet for measurement transfer.

30

Fig. 7 shows a standard IP (version 4) header and a standard UDP header. Checksum is not used (zero). Source and destination port are set to the lowest available port of the following (unassigned) range: 43191-44320.

35

The proprietary extension for transporting the measurements corresponds to the one explained with variant A.

5 Variant C:

The QoS PP is not implemented. The IP stack provides a direct or RAW IP socket as interface. In this case the same encapsulation as in Variant A is used. The proprietary transport protocol connects directly to the IP layer. Similarly, the TCP socket (not shown) could be used as interface.

Measurement request and Measurement transport

15 The measurement requests are sent from the AU collector to the sender in the modem. Measurements are sent from the sender in the modem to the AU collector. In both cases the same mechanism is used. The measurement request and the measurements are  
20 encapsulated in an IP packet with a proprietary transport protocol (AU collector implementation variants A and C, see above) or as proprietary payload in a UDP/IP packet with a special source and destination port (AU collector implementation B, see above). Two cases have to be distinguished:

25 1. An active PPP connection between the application unit and the modem is established (which means the modem is used and an IP connection to the network exists).

30 or

2. The modem is in idle mode. No PPP connection is active, no IP connection to the network is running.

35

## Case 1: Inbound packet transport

In this case the modem is used actually to transport IP packets using the modem over a (wireless) network. This means that an IP address was assigned to this modem connection. The sender within the modem generates IP packets with this assigned IP address as destination or recipient IP address; otherwise, they would be deleted. As sender or source IP address the next higher IP address is used, when the sender in the modem sends an IP packet to the application unit. The AU collector uses the assigned IP address of the modem connection as source address, when sending IP packets to the modem, and sets the next highest IP address as destination address. In the first measurement request from the AU collector the payload format is defined (implementation variant A, B, C, see above), in order for the modem to know, which variant to use.

NOTE: If the IP address in the application unit ends with .254, the next highest IP address is NOT .255 (broadcast address) but .1!

## Case 2: packet transport in IDLE mode

In an idle mode no IP address is assigned to the IP stack in the application unit for this modem connection. No PPP connection is active. If the AU collector wants to get measurements from the modem in idle mode it builds up a PPP connection to the modem using a very special string as dial up number (e.g. \*\*3\*4\*6\*\*# or \*\*f\*g\*m\*\*#), which is recognized from the sender in the modem as its own number. The IP address for this connection is assigned in the following way: The AU collector uses in the PAP (Password Authentication Protocol, part of the PPP initiation) as username a desired IP address for itself (it can be that other IP addresses are defined in the Applica-

tion unit already and the IP address of this connection must be unique). The modem assigns this IP address in the IPCP (IP Control Protocol) negotiation (which is part of the PPP negotiation) to the IP stack of the application unit for this modem connection. Again, the sender uses the next highest IP address for itself.

NOTE: If the IP address in the application unit ends with .254, the next highest IP address is NOT .255 (broadcast address) but .1!

If the modem should be used to connect to a wireless network this PPP connection is replaced and the system switches to case 1 operation.

#### Multidimensional decision matrix

The main controller 18 (see Fig. 1) or the decider (see Fig. 4), respectively, use a multidimensional decision matrix for dynamic packet flow or protocol optimization. Dynamic packet flow or protocol optimization and adaptation are more efficient than a static approach. Dynamic optimization and adaptation uses input parameters and events describing the quality or behavior of the underlying link as e.g.:

- Available bandwidth
- Used coding scheme (The coding scheme can be changed frequently during the connection, especially with the mobile EDGE standard.)
- Used timeslots, which vary a lot
- Bit Error Rate
- Cell Reselection
- Temporary loss of coverage

- RLC buffer status (generally data link layer stacks of the OSI reference model for mobile communication)
- etc.

- 5 Some of these input parameters may be predictions (coming from a state predictor), others may be "real-time" current measurements. For what follows, the difference between predicted and measured input parameters is not important.
- 10 Using these input parameters as input, the decider has to take a decision whether to trigger an action or to change some (protocol) parameters and which protocol parameters have to be changed to which value. Protocol parameters, in this context, refer to parameters of the layers 4 (transport layer) to 7
- 15 (application layer) of the OSI reference model for mobile communication, i. e. the higher layers. Examples for protocol parameters or actions are:
- Size of generated packets
  - Number of packets sent in one group
  - 20 - Frequency of packet generation
  - Start / stop of retransmission timer(s)
  - Triggering the transmission of a packet
  - Triggering the retransmission of a packets
  - Length of retransmission timer
  - 25 - Forward error correction
  - etc.

Not all actions are available at each time interval, because they are dependent on the current status of the protocol or

30 packet flow in the higher layers of the OSI reference model. It would not make any sense for instance to send out a new packet if the higher layer protocol stack is in the state to wait for an incoming acknowledgment for previously sent packets.

The goal of the decider is to find the optimum higher layer protocol behavior for the current (network) situation. It would be too time and processor capacity consuming, if the decider would have to make a deep analysis of the current situation. Using 4 or 5 input parameters describing the network situation and knowing all the possible states of the higher layer protocol stack easily brings up thousands of theoretical situations. To make the decision progress fast without losing accuracy of the decisions, a more dimensional decision matrix is used.

First of all, a list of all available actions for the higher layer protocol stack is generated. The decider cannot choose freely any action from this list. Which action has to be chosen strongly depends on the state of the higher layer protocol stack or, in other words, on previously taken actions. Therefore, for each higher layer protocol stack state a group of possible actions (which may include typically up to 4 actions) is created.

E.g. when sending a TCP/IP packet, the following states may be encountered:

1. state: wait for acknowledgement, retransmission timer not out of time,
2. state: IP packet sent, retransmission timer out of time,
3. state: acknowledgement received,
4. state: start of transmission of all IP packets (e. g. for an MMS),
5. state: end of transmission of all IP packets.

This is one dimension of the matrix with - here - 5 values associated to this first dimension.

The group of actions for state 3 is e.g.:

- a) do not change anything, send a packet with the same packet length and the same group size (e.g. 2 packets at a time) as before,
- 5    b) change the length of packets to half the length,
- c) change the length of packets to twice the length,
- d) increase the group size by one packet,
- e) decrease the group size by one packet,
- f) do not send anything, wait.

10

Now the input parameters for the decider are analyzed. They are reduced to the input parameters, which are most important for the optimization and adaptation process for this higher layer protocol.

15

For the above TCP/IP example, these are:

- prediction of a cell reselection,
- RLC buffer status, and
- available bandwidth.

20

This leads to three more dimensions of the matrix, making the matrix four-dimensional.

25

For the parameters "RLC buffer status" and "available bandwidth" a number of intervals (typically 3 or 4) are defined. The event, that an input parameter falls into a defined interval, is taken as input for the multidimensional decision matrix, leading to app. 4 input values for the three last dimensions of the matrix.

30

All in all, there are some hundred input parameter combinations for the matrix. For each input parameter combination a single action is defined.

One point in the matrix is, thus, described by one higher layer protocol state and the intervals for the most important input parameters. This matrix point has an action associated with it, which the decider has to trigger.

5

A change of action is only considered, if a change of an input value for the matrix is received. The new input parameter set marks a new point in the matrix. This point may have an action associated with it, which should be triggered by the decider.

10 However, there may be "empty" points in the matrix, where no action is required from the decider, if this point is reached.

While the present inventions have been described and illustrated in conjunction with a number of specific embodiments, those skilled in the art will appreciate that variations and modifications may be made without departing from the principles of the inventions as herein illustrated, as described and claimed. The present inventions may be embodied in other specific forms without departing from their spirit or essential characteristics. The described embodiments are considered in all respects to be illustrative and not restrictive. The scope of the inventions are, therefore, indicated by the appended claims, rather than by the foregoing description. All changes which come within the meaning and range of equivalence of the claims are to be embraced within their scope.

30